In the Claims:

Please amend the claims as follows:

1. (Currently Amended) A computer system, comprising:

multiple processors;

a plurality of resources assigned to node groups;

a first descriptor of respective topological levels of at least one of the resources, said first descriptor including an extended system descriptor; ~~and~~

said first descriptor having a pointer to a second descriptor of said resources, said ~~a~~ second descriptor including a node descriptor referenced in said first descriptor; ~~of respective performance of said resources,~~

said second descriptor including a node identifying number that identifies a path of interconnectivity of a resource in system topology, and a node number that identifies a node within said topology, said node identifying number and said node number being separate identifiers;

wherein the first and second descriptors are produced by a same firmware in a single computer system and said firmware is maintained as a data structure.

2. (Currently Amended) The system of claim 1, wherein said first descriptor is a first level data structure, and said second descriptor is a primary data structure.

3. (Original) The system of claim 2, wherein said primary data structure comprises a pointer to a secondary data structure.

4. (Currently Amended) The system of claim 1, wherein said firmware stores topology information of system resources in a data structure ~~further comprising a node identifier for each node for identifying~~

~~positional placement of a resource~~.

5. (Currently Amended) The system of claim 1 ~~4~~, wherein said node identifying number ~~identifier~~ is a string of multiple octets with a value stored in each octet identifying a location of a node ~~represents multiple levels of interconnect~~.

6. (Original) The system of claim 1, further comprising a dynamic updator of at least the first and second descriptors.

7. (Original) The system of claim 6, wherein said dynamic updator reflects real-time system configuration into the first descriptor.

8. (Original) The system of claim 6, wherein said dynamic updator reflects real-time system performance into the second descriptor.

9. (Currently Amended) The system of claim 1, wherein said second ~~first~~ descriptor includes a pointer to a secondary data structure having a descriptor selected from the group consisting of: processor descriptors, bus descriptors, memory descriptors, and share cache descriptors.

10. (Previously Presented) The system of claim 9, wherein said shared cache descriptor reflects interconnects of the system.

11. (Original) The system of claim 10, wherein said shared cache descriptor reflects latencies of the interconnects.

12. (Previously Presented) The system of claim 1, wherein said second descriptor reflects average latency between the node groups.

13. (Currently Amended) An article comprising:

a computer-readable recordable data storage medium readable by a computer having multiple processors and a plurality of resources assigned to node groups;

means in the medium for determining topological levels of at least some of the resources; and

means in the medium for determining performance of said resources,

wherein said topological level determining means and said performance determining means are capable of being stored in a same firmware of a single computer system, wherein said firmware is maintained as at least one data structure with a number stored in the data structure that identifies a path of interconnectivity of a resource within said topology.

14. Canceled

15. Canceled

16. (Original) The article of claim 13, wherein said topological level determining means is a first descriptor and said performance determining means is a second descriptor.

17. (Original) The article of claim 13, further comprising a node identifier for identifying positional placement of a resource for each node.

18. (Previously Presented) The article of claim 16, wherein said first descriptor includes a pointer to a secondary data structure having a descriptor selected from the group consisting of: processor descriptors, bus descriptors, memory descriptors, and share cache descriptors.

19. (Currently Amended) The article of claim 13, wherein ~~said~~ a shared cache descriptor reflects

interconnect of resources.

20. (Original) The article of claim 19, wherein said shared cache descriptor reflects latencies of the interconnects.

21. (Previously Presented) The article of claim 16, wherein said second descriptor reflects average latencies between node groups.

22. (Currently Amended) A method for enabling allocation of resources in a multiprocessor, comprising:

assigning multiple resources into node groups; and

maintaining system resource topology and performance descriptions as at least two ~~one~~ data structures ~~structure produce~~ produced by firmware in a single computer system, wherein at least one of said data structures includes an identifying number that identifies positional placement of a resource within said topology.

23. (Original) The method of claim 22, further comprising traversing the data structure to enable allocation of at least some of the resources.

24. (Original) The method of claim 22, wherein said traversal step includes accessing a second data structure.

25. (Previously Presented) The method of claim 24, wherein said second data structure is selected from the group consisting of: processor descriptors, bus descriptors, memory descriptors and shared cache descriptors.

26. (Previously Presented) The method of claim 24, wherein said second data structure includes a shared cache descriptor for describing at least part of a system interconnect including latency between sibling nodes.

27. (Original) The method of claim 22, further comprising maintaining at least average latency between at least two of the nodes.

28. (Original) The method of claim 22, wherein said traversal step includes recursively accessing additional data structure levels.

29. (New) The article of claim 13, wherein said identifying number is a string of multiple octets with a value stored in each octet identifying a location of a node.

30. (New) The method of claim 22, wherein said identifying number is a string of multiple octets with a value stored in each octet identifying a location of a node.